III International Conference on Particle-based Methods – Fundamentals and Applications
PARTICLES 2013
M. Bischoff, E. Oñate, D.R.J. Owen, E. Ramm & P. Wriggers (Eds)

A DYNAMIC ADAPTATION TECHNIQUE FOR THE MATERIAL POINT METHOD

K. P. Ruggirello, and S. C. Schumacher[1]

[1]Sandia National Laboratories
PO Box 5800
Albuquerque, NM 87185-0828
kruggir@sandia.gov

Key words: Material point method, MPM, CPDI, adaptive

Abstract. The Lagrangian Material Point Method (MPM) [1, 2] has been implemented into the Eulerian shock physics code CTH[3], at Sandia National Laboratories. Since the MPM uses a background grid to calculate gradients, the method can numerically fracture if an insufficient number of particles per cell are used in high strain problems. Numerical fracture happens when the particles become separated by more than a grid cell leading to a loss of communication between them. One solution to this problem is the Convected Particle Domain Interpolation (CPDI) technique[4] where the shape functions are allowed to stretch smoothly across multiple grid cells, which alleviates this issue but introduces difficulties for parallelization because the particle domains can become non-local. This paper presents an approach where the particles are dynamically split when the volumetric strain for a particle becomes greater than a set limit so that the particle domain is always local, and presents an application to a large strain problem.

1 INTRODUCTION

Eulerian hydrodynamic methods are useful for large deformation problems, where mesh tangling typically leads to difficulties for Finite Element methods. However, Eulerian techniques suffer from diffusion due to numerical advection errors, which can be problematic for many material models requiring the transport of history variables that should remain sharp[5]. The MPM alleviates the issue of numerical diffusion in Eulerian methods, and also does not suffer from mesh tangling during large deformation and fragmentation.

The original MPM has been shown to suffer from cell crossing noise due to the lack of smoothness of the shape functions, and numerical fracture when the particles become separated by more than a grid cell. To alleviate the cell crossing noise Bardenhagen et al.[6] introduced the Generalized Interpolation Material Point (GIMP) method, which allows the particle domains to deform aligned with the background grid and constructs smooth shape functions. However the GIMP method does not capture the rotation of

the particle domains. Sadeghirad et al.[4] introduced the CPDI method which allows the particle domains to deform as parallelograms and constructs smooth basis functions on the grid. The CPDI method has been shown to simulate large deformation problems accurately and alleviates the cell crossing noise present in the standard MPM.

Since the GIMP and CPDI methods allow the particle domains to stretch across multiple cells, their basis functions can have non-local support during massive deformation problems which can lead to implementation issues in existing codes and with parallelization. This paper presents a splitting algorithm for the CPDI method which allows a particle to be split when it becomes too highly deformed. The splitting of particles is also desirable in regions of high deformation so that additional degrees of freedom can be introduced dynamically in the problem where needed, analogous to adaptive mesh refinement (AMR) used in Eulerian codes such as CTH[3]. Using grid based AMR in conjunction with the MPM also necessitates the splitting of particles if the grid is extensively refined so that the particles do not numerically fracture.

Previously Giovanni and Brackbill[7] presented a particle splitting approach for the particle-in-cell (PIC) method which focused on enforcing the constraint that the data on the background grid was consistent between the unsplit and split particle configurations. This resulted in an overdetermined linear algebraic system which needed to be solved via Lagrange multipliers for each cell with particles to be split in it. Tan and Nairn[8] developed an algorithm for splitting particles in the MPM method to capture dynamic energy release from cracks. Shang et al.[9] also presented a method for splitting particles using the MPM and applied it to the simulation of shaped charges.

The remainder of this paper will be organized as follows. First the particle splitting algorithm will be introduced, followed by its application to a problem involving massive deformations. Finally conclusions will then be drawn from the results. For an overview of MPM and the CPDI method the reader is referred to Sulsky et al.[1], and Sadeghirad et al.[4], respectively.

2 PARTICLE SPLITTING ALGORITHM

In the CPDI method, a particle is initially defined as a parallelogram which is deformed during the calculation. The deformation is tracked through the temporal integration of the deformation gradient tensor, $\underset{\sim}{\boldsymbol{F}}$, which maps the undeformed particle domain to the deformed one. Figure 1(a) shows an initial particle domain dimensioned by $\boldsymbol{r}_1^0$ and $\boldsymbol{r}_2^0$, and Fig. 1(b) shows the updated parallelogram particle domain at time step $n+1$. The particle domain is determined according to,

$$\begin{aligned} \boldsymbol{r}_1^{n+1} &= \underset{\sim}{\boldsymbol{F}}_p^{n+1} \boldsymbol{r}_1^0 \\ \boldsymbol{r}_2^{n+1} &= \underset{\sim}{\boldsymbol{F}}_p^{n+1} \boldsymbol{r}_2^0 \end{aligned} \tag{1}$$

The basis functions are evaluated at the particle domain corners, allowing the particle to stay connected to its neighbors even during large strains where it can potentially span

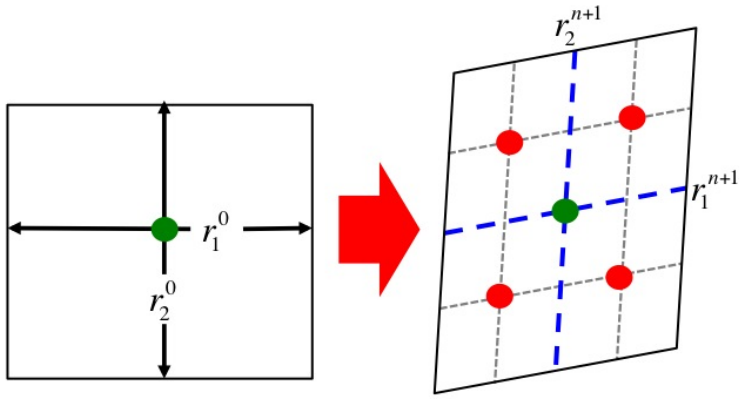


Figure 1: Initial and deformed CPDI particle domain being split in 2D.

multiple cells.

The splitting algorithm is only performed every 10 cycles for this study to reduce computational time, the results presented are found to not be sensitive to this number. The first step in the particle splitting algorithm is to determine which particles to split. For this study the criteria for determining if a particle is split was when the Jacobian becomes greater than a set tolerance,

$$J = \det \underset{\sim}{\boldsymbol{F}} \geqslant \epsilon \tag{2}$$

This serves to split the particle if the volume of the particle domain becomes larger than ϵV^0, where V^0 is the initial particle domain volume. Another criteria that could have been used is to check if the distance or number of cells the particle domain spans in any direction becomes greater than a set tolerance. However this criteria would have been more computationally expensive and the criteria used in this study is found to be sufficient for problems of interest.

The center for the split particles are placed at,

$$\begin{aligned} x_{sp} &= x_p \pm \frac{1}{2} \underset{\sim}{\boldsymbol{F}}_p \boldsymbol{r}_1^0 \\ y_{sp} &= y_p \pm \frac{1}{2} \underset{\sim}{\boldsymbol{F}}_p \boldsymbol{r}_2^0 \end{aligned} \tag{3}$$

where subscript sp refers to the split particle and p to the original particle. Figure 1 (b) shows the deformed particle being split into 4 particles in 2D with their new center locations denoted in green. The placing of the new particles at the quadrants of the deformed particle is similar to the approach of Tan and Nairn[8] where the split particles were placed accounting for the current deformation field of the original particle. The mass for the split particles is defined as $m_{sp} = m_p/4$, while the energy, temperature, pressure,

stress, strain, deformation gradient, velocity, and history variables are set equal to the original particle. The original particle is removed after the split particles are added. The algorithm can easily be extended to a 3D geometry.

3 RESULTS

Conservation errors resulting from the splitting algorithm outlined in Sec. 2 were quantified by taking a constant velocity object in void and advecting it through the simulation domain while splitting it every 100 computational cycles, the simulation was run for 1000 cycles resulting in 10 levels of splitting. Table 1 summarizes the maximum percent conservation errors for the problem and shows they are all within the range of round-off error.

Table 1: Maximum Conservation errors for 10 levels of particle splitting.

Property	% Error
Mass	$1.1293E-15$
Momentum	$1.0473E-14$
Energy	$1.0132E-14$

3.1 EXAMPLE PROBLEM

The splitting of particles is desired for problems involving massive deformation in order to introduce additional degrees of freedom in the solution. An application which is frequently encountered is the modeling of shaped charges where the jet formed exhibits massive straining. For this paper a surrogate shaped charge is modeled as shown in Fig. 2, where 2 copper places perpendicular to each other are given a normal velocity of $1.2km/s$ with a domain of $10cm \times 40cm$. The Johnson-Cook[10] material model, and the Mie-Gruneisen equation of state is used for the copper. The criteria for splitting particles is chosen to be $\epsilon = 1.5$.

A mesh of 75×300 is used in the simulation with 16 particles per cell initially and 3 levels of refinement. All boundaries conditions of the domain are open. The initial number of particles is 4600 and 47019 by the end of the simulation, showing a large amount of splitting took place in the problem as expected. Figure 3 (a-c) shows the simulation results at $19\mu s$, $38\mu s$, and $88\mu s$, respectively. The jet from the surrogate shaped charge is accurately captured even with the large deformation.

A useful measurement for shaped charges is the tip velocity of the jet. Table 2 shows the resulting jet tip velocity and run times for the surrogate shaped charge with no particle adaption, 2 levels of splitting, 3 levels of splitting, and a high resolution mesh converged case. The high resolution case was run at a mesh resolution of 75×300 with 256 particles per cell and no particle adaption, the initial number of particles in this case is equivalent

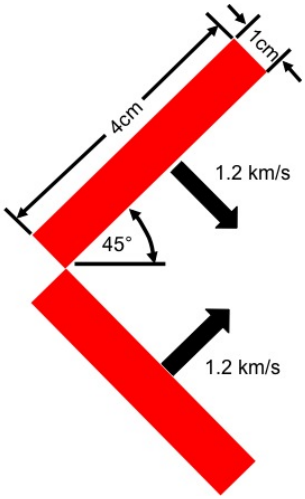
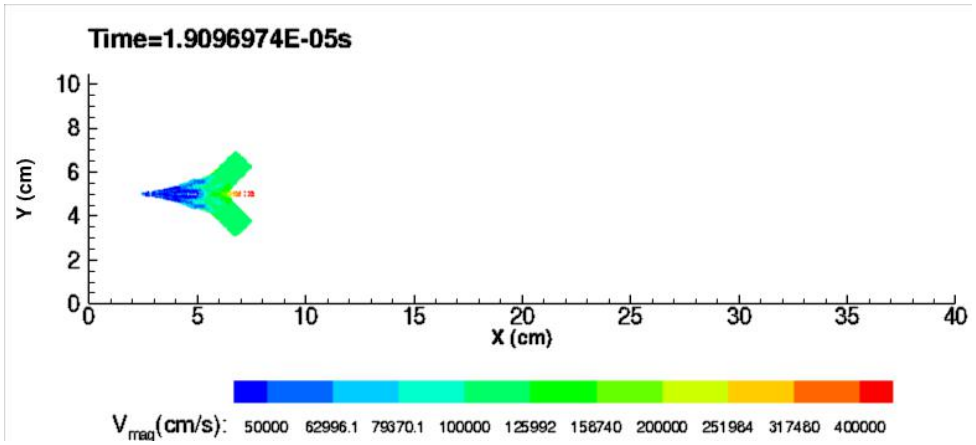


Figure 2: Example problem setup.

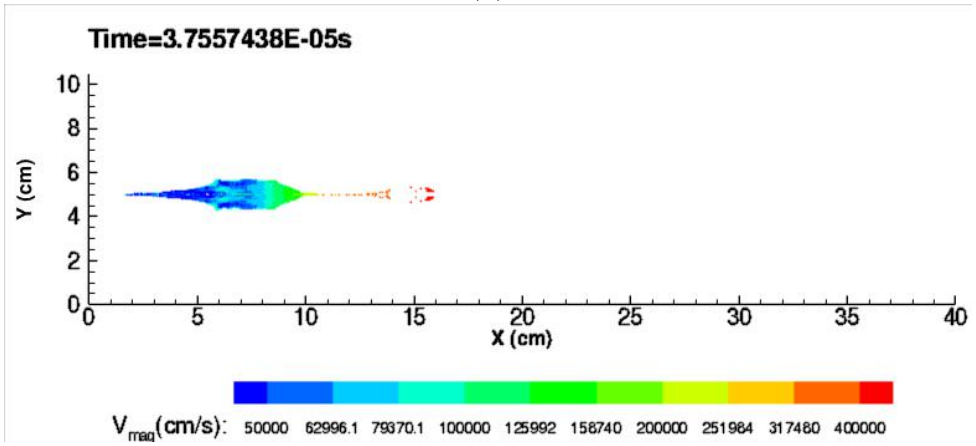
to the case with 3 levels of adaptation if the entire problem is refined 3 levels. The high number of grid cells and particles per cell is needed in order to accurately resolve the large stretching in the jet. With no adaptation the tip velocity is under predicted by 8.8%, and as the number of levels of particle adaptation is increased the tip velocity converges to the high resolution case. With 3 levels of splitting the total run time is approximately 28% less than the run time for the mesh converged case, and the tip velocity is within 0.65%. Compared to the case with no adaptation, 3 levels of adaptation takes approximately four times more computational time due to the additional complexities of the algorithm and large number of particles generated. For problems where the majority of the problem undergoes massive deformation, particle splitting is attractive for reducing the run time while still retaining accuracy.

Table 2: Tip velocity and run times for surrogate shaped charge for no adaption, adaption, and high resolution cases.

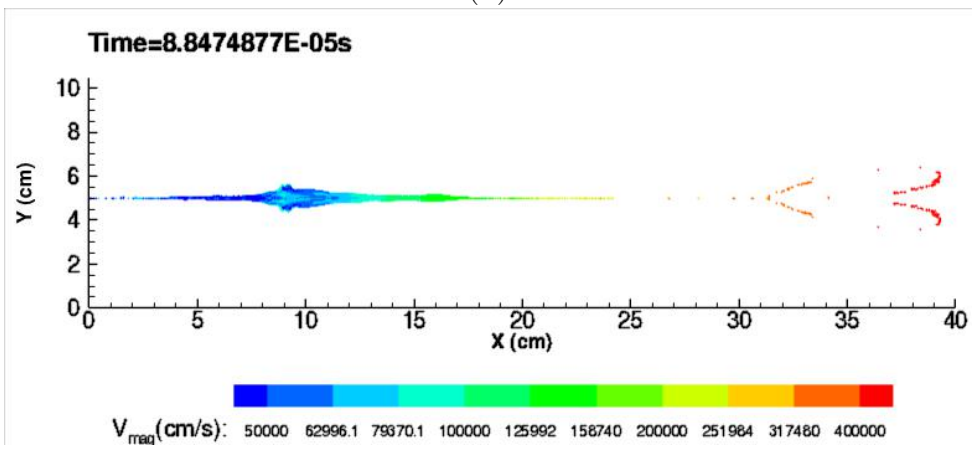
	Jet Tip Velocity (km/s)	Run Time (s)	% Error
No Adaptation	3.40124	40.2	8.8
2 Levels of Adaptation	3.60984	132.1	3.2
3 Levels of Adaptation	3.70457	160.2	0.65
Mesh Converged Case	3.72862	224.1	0



(a)



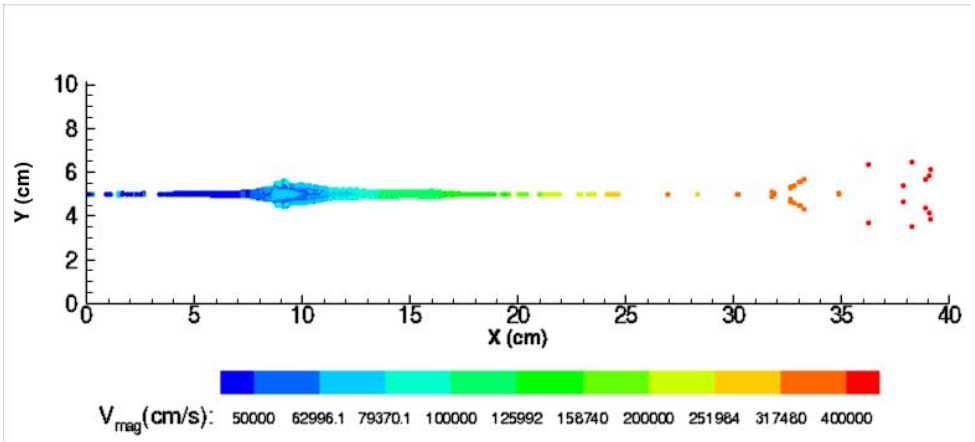
(b)



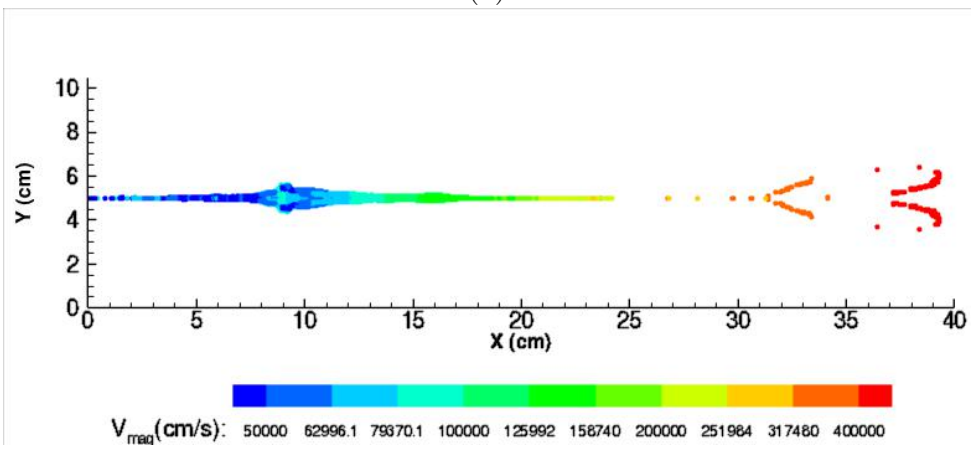
(c)

Figure 3: Representative results for surrogate shaped charge at times of (a) 19μs, (b) 38μs, and (c) 88μs using particle adaption.

Figure 4 (a) and (b) show the final results of the surrogate shaped charge at 0.9e-5s without and with 3 levels of particle splitting, respectively. The results are qualitatively similar, showing that particle splitting did not introduce any spurious artifacts in the solution. The position of the jet tip with no splitting is offset slightly from the case with splitting since the tip velocity is under predicted with no adaptation. Figure 5 (a) and (b) show an enlarged view of the jet at 0.9e-5s without and with particle splitting, respectively. Without adaptation there are large areas where there is insufficient resolution in the jet, but with adaptation additional degrees of freedom are automatically added in this area and the jet is well resolved.

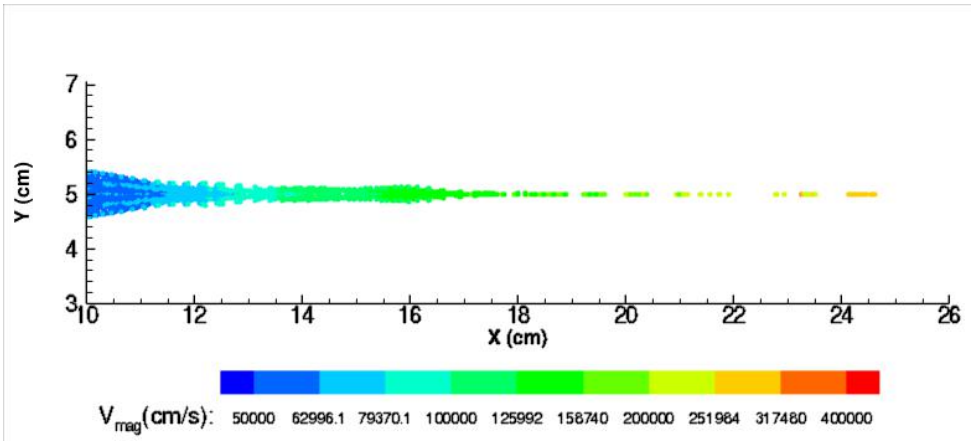


(a)

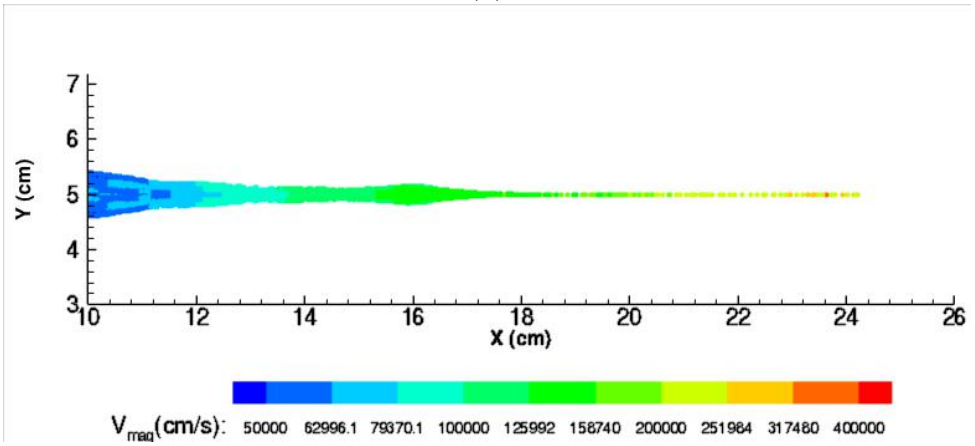


(b)

Figure 4: Comparison of non-adaptive (a) and adaptive (b) results at a time of 0.9e-5s for the surrogate shaped charge.



(a)



(b)

Figure 5: Comparison of non-adaptive (a) and adaptive (b) results at a time of 0.9e-5s for the surrogate shaped charge looking at jet.

4 CONCLUSIONS

An algorithm for splitting particles and splitting criteria in the CPDI method while retaining mass, momentum, and energy conservation is presented. The method is shown to not introduce any spurious artifacts in the solution and automatically introduces additional degrees of freedom in areas where it is desired. The overall run time of the code with 3 levels of splitting is approximately 28% less than running the same problem with the entire problem refined to an equivalent number of particles.

REFERENCES

[1] D. Sulsky, Z. Chen, and H. Schreyer, "A particle method for history-dependent materials," *Computer Methods in Applied Mechanics and Engineering*, vol. 118, pp. 179–196, Sept. 1994.

[2] S. C. Schumacher, K. P. Ruggirello, and B. A. Kashiwa, "CTH marker lagrangian capabilities," in *SAVE Conference*, (New Orleans, LA), Nov. 2012.

[3] J. McGlaun, S. Thompson, and M. Elrick, "CTH: a three-dimensional shock wave physics code," *International Journal of Impact Engineering*, vol. 10, no. 14, pp. 351–360, 1990.

[4] A. Sadeghirad, R. M. Brannon, and J. Burghardt, "A convected particle domain interpolation technique to extend applicability of the material point method for problems involving massive deformations," *International Journal for Numerical Methods in Engineering*, vol. 86, pp. 1435–1456, 2011.

[5] S. C. Schumacher, C. W. S. Bruner, E. N. Harstad, and C. T. Key, "Modeling layered composite materials using MPM in an eulerian shock physics code," in *15th European Conference on Composite Materials*, (Venice, Italy), June 2012.

[6] S. Bardenhagen and E. Kober, "The generalized interpolation material point method," *CMES: Comput. Model. Engrg. Sci.*, vol. 5, pp. 477–495, 2004.

[7] L. Giovanni and J. U. Brackbill, "Control of the number of particles in fluid and MHD particle in cell methods," *Computer Physics Communications*, vol. 87, pp. 139–154, 1995.

[8] H. Tan and J. Nairn, "Hierarchical, adaptive, material point method for dynamic energy release rate calculations," *Comput. Methods Appl. Mech. Engrg.*, vol. 191, pp. 2095–2109, 2002.

[9] M. Shang, X. Zhang, L. Yanping, and X. Zhou, "Simulation of high explosive explosion using adaptive material point method," *CMES: Comput. Model. Engrg. Sci.*, vol. 39, no. 2, pp. 101–123, 2009.

[10] G. R. Johnson and T. J. Holmquist, "Evaluation of cylinder-impact test data for constitutive model constants," *J. Appl. Phys.*, vol. 64, no. 8, pp. 3901–3910, 1988.

5 ACKNOWLEDGEMENTS

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.